RECEIVED
CENTRAL FAX CENTER
AUG 09 2006



## REMARKS

Applicants amend claims 1, 12, 24, 32, 34, 43, 52, 61, 66, 70, and 74 to clarify the claimed invention. No new matter is added. Applicants respectfully submit that the pending claims define over the art of record.

Objection to the Specification

The Examiner objects to the incorporation by reference of the Appendix in the specification. Applicants amend the specification as suggested by the Examiner. Applicants respectfully submit that the material being inserted in the specification is the material previously contained in the Appendix and incorporated by reference and that the amendment contains no new matter. Please delete the Appendix section from the Specification. Applicants respectfully request that the Examiner reconsider and withdraw the objection to the specification.

Claim Rejections Under 35 U.S.C. §112

Claims 1, 12, 24, 32, 61, 66, 70, and 74 and their corresponding dependent claims stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite due to the phrase "in conjunction with." Applicants amend the claims to address the Examiner's concerns. Applicants respectfully request that the Examiner reconsider and withdraw the rejection of claims under 35 U.S.C. §112, second paragraph.

Summary of the Claimed Invention

This application is directed to using graphically represented functions within a finite state machine—that is, within a model of a system represented as a finite state machine. The graphical function may be called from one or more states or transitions in the finite state machine. A single graphical function may be called multiple times from within different places in the finite state machine.

The finite state machine can use the graphically represented function by textually referencing the graphical function. In other words, the graphical function can be called textually,

by its name. This is advantageous because the entire graphical function does not need to be repeated graphically at every point of call in the executable model.

A graphical function can be defined by any diagram that can describe a function, such as, for example, by elements from data flow diagrams, control flow diagrams, and/or state diagrams.

Claim Rejections Under 35 U.S.C. §102

Claims 1-18 and 20-78 are rejected under 35 U.S.C. §102(e) as being anticipated by United States Patent Application Publication No. 2002/0083413 to Kodosky et al. (hereafter "Kodosky"). This rejection if respectfully traversed and reconsideration is requested. Applicants respectfully submit that the Kodosky reference does not disclose each and every element of the pending claims.

Claims 1-6 and 24-33

Claim 1, 24, and 32 recite, among other things, that the graphical representation is an executable model of the finite state machine and the function that is represented graphically is called from within the graphical representation of the finite state machine. Applicants respectfully submit that the Kodosky reference does not disclose these limitations.

Kodosky discloses in Fig. 19 a side by side comparison of a state diagram and a graphical program. The state diagram is shown at the right-hand side of Fig. 19 and it is an input to an application to generate the graphical program, which is shown at the left-hand side of Fig. 19. The state diagram is a conventional diagram that itself cannot be executed or simulated. The graphical program is implemented using a graphical programming language, and it is a code representation of a state diagram. In other words, Kodosky merely teaches how to translate a non-executable state diagram to code so that it can be executed. The graphical program includes graphical code that mimic the behavior shown in the state diagram. The graphical program does not make a function call to the state diagram because the state diagram is not used to represent a function and the state diagram cannot be executed like a function.

In contrast, the claimed invention requires a graphical representation of a finite state machine that is executable and a function that can be presented graphically and be called by the

graphical representation of the finite state machine and that is also executable. That is, there are at least two executable entities: the executable model of the finite state machine and the graphical function called from within the finite state machine. Kodosky has at most one executable entity—the code which is derived from the state diagram. Kodosky does not teach or suggest calling a graphical function from within the executable model of the finite state machine.

Kodosky discloses in paragraph 166, lines 11-15 that a state diagram editor may support execution highlighting. Execution highlighting is a feature of the Kodosky's state diagram editor and not a function being called by a graphical representation of a finite state machine. Moreover, Kodosky discloses generation of a graphical program from a state diagram using a graphical programming environment, where the graphical program can be compiled or interpreted to produce machine language. However, the graphical program of Kodosky is not a graphical representation of a finite state machine that makes a call to a graphical function by referencing. Even if the graphical program of Kodosky were to be viewed as a graphical representation of a finite state machine, *arguendo*, Kodosky does not teach both a graphical representation of a finite state machine *and* a graphical representation of a function that is used by the graphical representation of the finite state machine, as required by claims 1, 24, and 32. Although a graphical function may be a state diagram, the claim language of claims 1, 24, and 32 would require the Examiner to show two separate state diagrams, one as a graphical representation of a finite state machine and the other as a graphical representation of a function that is called from within the finite state machine.

Accordingly, Applicants respectfully submit that Kodosky fails to disclose the each and every element and limitation of amended claims 1, 24, and 32. Applicants respectfully request that the Examiner reconsider and withdraw the rejection of claims 1, 24, and 32 and their corresponding dependent claims.

Claims 7-18 and 20-23

Independent claims 7, 12, and 17 recite the limitation of receiving user input defining at least one graphical function. Applicants respectfully submit that the Kodosky reference fails to disclose this limitation. Kodosky discloses at paragraph 63, lines 1-8 that a program can receive state diagram information and programmatically generates a graphical program based on the

state diagram. The state diagram information is not a user input, and even if it can be considered as one, the state diagram has already been analogized by the Examiner to the finite state machine recited in the independent claims 7, 12, and 17. The same element of Kodosky cannot both be the finite state machine from within which a graphical function is called and the graphical function which is being called. Accordingly, Applicants respectfully submit that the Kodosky reference fails to disclose each and every element and limitation of claims 7, 12, and 17. Applicants respectfully request that the Examiner reconsider and withdraw the rejection of independent claims 7, 12, and 17.

Applicants note that the dependent claims also recite separate patentable subject matter. For example, claims 10, 15, and 21 recite the limitation that the user input comprises a function flow diagram. Kodosky merely discloses a state diagram can be an input to a program or a user can enter manual textual source code into a graphical program. Nowhere does the Kodosky reference disclose the limitation that user input can include a function flow diagram, as recited in claims 10, 15, and 21. As such, for this and the reasons set forth above, the dependent claims also define over the art of record.

Claims 34-60

Independent claims 34, 43, and 52 are amended to recite limitation of a component (or block component) representing a function and reference by at least one of the states or at least one of the transitions to call the function at one of the states or one of the transitions and the limitation of providing one or more block components representing one or more states in an executable model. Applicants respectfully submit that the Kodosky reference does not disclose these limitations.

Kodosky discloses at paragraph 17, lines 1-4 that each state may represent some instructions or sequence of instruction that is executed when the state is active. However, Kodosky does not disclose a component representing a function and *referenced by* a state or a transition *to call the function within the state or the transition*, as required by independent claims 34, 43, and 52. Hence, the Kodosky reference cannot have the advantage of the claimed invention that allows a function to be defined once and called later to perform the same

instructions without the need to repeat the same set of instructions multiple locations in a state diagram.

Kodosky discloses in Fig. 1 a state diagram including states and transitions. However, the state diagram is not executable. The state diagram is merely used as an input in the Kodosky reference to generate a graphical program. In contrast, the claimed invention requires that the one or more block components are provided in an executable model, where the block components represent one or more states.

Accordingly, Applicants respectfully submit that the Kodosky reference fails to teach each and every element and limitations of independent claims 34, 43, and 52. Applicants respectfully request that the Examiner reconsider and withdraw the rejection of claims 34, 43, and 52.

Applicants note that the dependent claims also recite separate patentable subject matter. For example, claims 42, 51, and 60 recite the limitation of a shadowing function, wherein the shadowing function comprises using in a function invocation a function definition proximally closest to a point of invocation of the function in a state diagram hierarchy. The claimed invention hence has the capability of defining multiple functions using the same name and when a function of such name is invoked at a certain location, the function definition that is defined closest to location of invocation will be used. The Kodosky reference merely discloses in paragraph 20 that priority can be assigned to different transitions to ensure only one transition becomes active. Nowhere does the Kodosky reference disclose a shadowing function that includes using a function invocation a function definition proximally closest to a point of invocation of the function. The Kodosky reference also does not disclose assigning priority orders to different function definitions. Even if the Kodosky reference teaches assigning priority orders to different function definitions, the function with the highest priority will always be invoked, regardless of the location of the function with the highest priority. Hence, the Kodosky reference cannot teach a function invocation a function definition proximally closest to a point of invocation of the function in a state diagram hierarchy. As such, for this and the reasons set forth above, the dependent claims also define over the art of record.

Claims 61-78

Independent claims 61, 66, 70, and 74 are amended to recite the limitation of textually referencing the graphically represented function within the model to cause an invocation of the graphically represented function during execution of the model. Applicants respectfully submit that Kodosky does not disclose this limitation.

Kodosky discloses in paragraph 33, a GPG program can be invoked to receive state diagram information to generate a graphical program. The state diagram information may be in many different formats, such as binary data, XML data, or text data. In other words, Kodosky discloses an application program that can translate a state diagram to a graphical program. However, Kodosky does not disclose *referencing a graphically represented function textually in an executable model* to cause an invocation of the graphically represented function during execution of the model. The state diagram discloses by Kodosky is not an executable model or a graphically represented function that can be referenced to textually in an executable model. The state diagram in Kodosky is merely an input to an application program.

Accordingly, Applicants respectfully submit that the Kodosky reference does not disclose each and every element and limitation of claims 61, 66, 70, and 74. Applicants respectfully request that the Examiner reconsider and withdraw the rejection of claims 61, 66, 70, and 74 and their corresponding dependent claims.

Application No.: 09/855199 Docket No.: MWS-070RCE

## CONCLUSION

In view of the above amendment, Applicants believe the pending application is in condition for allowance.

Applicants believe no fee is due with this statement. However, if a fee is due, please charge our Deposit Account No. 12-0080, under Order No. MWS-070RCE from which the undersigned is authorized to draw.

Dated: August 9, 2006

Respectfully submitted,

By _____

Weijen Wang
Registration No.: L0257
LAHIVE & COCKFIELD, LLP
28 State Street
Boston, Massachusetts 02109
(617) 227-7400
(617) 742-4214 (Fax)
Agent For Applicant